CS 590 NLP

HW2

Edit Distance

Due 02/02 11:59 pm

In this homework, you will be creating a simple spell checker program in python which uses the dictionary you created in HW1. This program will function as follows (note the '>>>' indicate actions from the python shell): (Note that your program may not give the exact same suggestions).

HW1:

```
>>> process_regex("warofworlds.txt")
Processing file...
Output stored to "regex.txt"

>>> normalize_text("regex.txt")
Normalizing text...
Output stored to "dictionary.txt"
```

HW2:

```
>>> spell_checker()
----------------------------------------
Welcome to the spell checker!
Please enter a text to check spelling or enter quit to exit the program.
----------------------------------------
Enter text to be checked: The rain in Spain, falls mainly on the plain.
No misspellings detected!

Enter text to be checked: The raen in Spain, fals mainly on the plain.
Misspelling - Suggestion
------------------------------
raen – rain
fals - falls

Enter text to be checked: quit
Goodbye!

>>>
```

Spell Checker

Your spell checker will follow the following rules:

1. If a word in the input text is not in the dictionary it is considered a misspelled word.
    a. Note you must avoid thinking that punctuation indicates misspelling (e.g. Spain,)
2. If a word is misspelled, then the spell checker should find the closest word to it by leveraging the minimum edit distance formula discussed in class. Specifically, the word in the dictionary which has the smallest minimum edit distance to the misspelled word should be suggested.
    a. (You may try some optimizations such as storing already calculated edit distances, or only checking a subset of dictionary words, if you want to speed your program up)
3. The spell checker should be called by a function called "**spell_checker()"**
4. Note that you may have multiple "closest" words. It is up you to whether you just randomly choose one of those words or present all of those words. (e.g. raen – rain, ran, rein)

Report

Test out your spell checker with different phrases and misspellings of your choice. Put these in your report as well as the suggestions output by your program.

Write out some discussion in your report on the tests you ran. Are there obvious problems with the suggestions? Are there good examples where edit distance is working as a spell detector? Note any problems you encountered along the way.

Additional Rules (MUST BE FOLLOWED):

1. The code should be written in python 3.
2. Standard libraries should only be used for this assignment, ie. I shouldn't have to download extra libraries to run your code.
3. You should make your code modular to the different steps. (e.g. **at least one function for the spell check interaction step**). (You probably will have more functions to help your main functions)
4. You should only hand in one python file: **USERNAME_HW2.py**. And your dictionary file.
    a. Note: no jupyter files.
5. You should be adding comments to document and communicate your thought process. **If I can't understand why you perform an action, then I can't credit you for performing that action.**

Grading

Assignment will be graded as follows:

Description	Points
Code Runs	5
Spell Checker Implementation/Correctness	20

Outputs follow rules	5
Documentation (Comments, functions, etc)	5
Report	15
Total:	**50**

- **If the code does not run, I cannot grade it well.** (More points than 5 can be lost if the code cannot be run, as I will not be able to fully test the implementations of the other functions).
- **Breaking of the additional rules can result in applied penalties.** (Always make sure you are checking against the rules)

Suggestions

- **Documentation is key for showing your effort in this homework.** Make sure you are noting why you make certain decisions all throughout your code.
- The slides for previous classes are posted, so please refer to these and the book for ideas during implementation.
- Start simple, build up complexity. You should always make sure your new ideas being added do not cause your program to crash. So starting simple is the best way to a) maintain the ability to keep your code running, b) add in comments for documentation and thought process as you add more code.
- Work through the homework yourself, rather than sharing ideas (especially not code) with other students. **As a reminder, plagiarism (or sharing) of code is strictly prohibited.** This assignment is complex enough that significant overlap between students will be suspicious.
- If you have not worked with python before, w3schools can help you translate your previous coding experience to python (https://www.w3schools.com/python/default.asp)
- Stop by office hours to discuss ideas. I am always happy to help you think through your process!